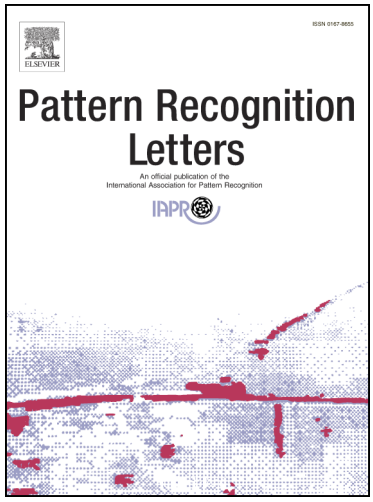
Adaptive conformal semi-supervised vector quantization for dissimilarity data

Zhu, Xibin; Schleif, Frank-michael; Hammer, Barbara

DOI:
10.1016/j.patrec.2014.07.009



Adaptive Conformal Semi-Supervised Vector Quantization for Dissimilarity Data

Xibin Zhu, Frank-Michael Schleif, Barbara Hammer

PII: S0167-8655(14)00226-8
DOI: http://dx.doi.org/10.1016/j.patrec.2014.07.009
Reference: PATREC 6057

To appear in: *Pattern Recognition Letters*

Please cite this article as: Zhu, X., Schleif, F-M., Hammer, B., Adaptive Conformal Semi-Supervised Vector Quantization for Dissimilarity Data, *Pattern Recognition Letters* (2014), doi: http://dx.doi.org/10.1016/j.patrec.2014.07.009

Pattern Recognition Letters
journal homepage: www.elsevier.com

Adaptive Conformal Semi-Supervised Vector Quantization for Dissimilarity Data

Xibin Zhu[a,**], Frank-Michael Schleif[b], Barbara Hammer[a]

[a]*Bielefeld University, Center of Excellence, Inspiration 1, 33619 Bielefeld, Germany*
[b]*School of Computer Science, University of Birmingham, Birmingham B15 2TT, UK*

ARTICLE INFO

Article history:

Semi-Supervised Learning
Proximity Data
Dissimilarity Data
Conformal Prediction
Generalized Learning Vector Quantization

ABSTRACT

Existing semi-supervised learning algorithms focus on vectorial data given in Euclidean space. But many real life data are non-metric, given as (dis-)similarities which are not widely addressed. We propose a conformal prototype-based classifier for dissimilarity data to semi-supervised tasks. A 'secure region' of unlabeled data is identified to improve the trained model based on labeled data and to adapt the model complexity. The new approach (i) can directly deal with arbitrary symmetric dissimilarity matrices, (ii) offers intuitive classification by sparse prototypes, (iii) adapts the model complexity. Experiments confirm the effectiveness of our approach in comparison to state-of-the-art methods.

1. Introduction

Due to the sheer amount of data, only few of these data are completely labeled, and labeling of all data is indeed very costly and time consuming. Accordingly many data sets, in life sciences for example, are only partially labeled. Techniques of data mining, visualization, and machine learning are necessary to help people to analyze those data. Especially semi-supervised learning (SSL) techniques are widely used for this setting. The idea of semi-supervised learning is to learn the model not only from the labeled training data, but to also incorporate structural and statistical information in additionally available unlabeled data. A variety of SSL methods has been published (Zhu and Goldberg, 2009). Most of them focus on vectorial data given in Euclidean space or representations by means of positive semi-definite (psd) kernel matrices.

A lot of real world data, like biological sequences, are non-vectorial, often non-Euclidean and given in the form of pairwise proximities, which are based on pairwise comparisons of objects providing some score-value of the (dis-)similarity of the objects. Those data are also referred to as *proximity* or *relational data*. An underlying vector space is not necessarily available and there is no guarantee of metric conditions. Examples of those proximity or (dis-)similarity measures are edit distance based measures for strings or images (Haasdonk and Bahlmann, 2004) or popular similarity measures in bioinformatics such as scores obtained by the Smith-Waterman, FASTA, or blast algorithm (Gusfield, 1997).

Methods based on partially labeled similarity data, where the similarities are defined on a metric space, as discussed in (Pekalska and Duin, 2005), can be effectively handled by semi-supervised extensions of kernel methods or other recently proposed, effective strategies (Subramanya and Bilmes, 2011; Tanha et al., 2014). However, in case of non-metric (dis-)similarity data without an explicit vector representation and without requesting a metric space only few methods have been proposed so far in the literature of SSL (Trosset et al., 2008), and kernel techniques can be applied using some costly, potentially degenerating, transformations on the proximity data only (Pekalska and Duin, 2005).

First, we take a glance at SSL methods. One way to categorize SSL methods is to divide the field into generative models, low-density separation methods, and graph-based techniques typically used for a classification objective. A recent introduction to SSL is given in (Zhu and Goldberg, 2009). In generative models, the most basic technique is given by *self-training*. A classifier is first trained on the labeled instances and is then applied to unlabeled instances. Usually, some subset of those

**Corresponding author
e-mail: xzhu@techfak.uni-bielefeld.de (Xibin Zhu), schleify@cs.bham.ac.uk (Frank-Michael Schleif), bhammer@techfak.uni-bielefeld.de (Barbara Hammer)

newly labeled instances are then used together with the original labeled data, to retrain the model. The major advantages of self-training are its simplicity and the fact that it is a wrapper method. It can 'wrap' the learner without changing its inner workings. In this paper we adopt this approach.

Besides EM-based methods (e.g. (Suzuki et al., 2007)) and graph-based techniques (e.g. (Zhu and Goldberg, 2009; Zhang et al., 2014)), probably the most popular semi-supervised learner in low-density separation methods is the transductive Support Vector Machine (TSVM) or its variants (Chapelle et al., 2006). The semi-supevised SVM (S3VM) aims at approaching one optimal low-density separator employing unlabeled data, whereas *Safe* S3VM (S4VM) (Li and Zhou, 2011) tries to exploit multiple candidate low-density separators simultaneously to reduce the risk of identifying a poor separator with unlabeled data. Besides, multi-kernel approaches have been recently analyzed for S3VM to incorporate additional meta-knowledge in the semi-supervised optimization (Tian et al., 2012). While most of these methods are defined for two-class problems, employing e.g. one-vs-rest wrappers for the multi-class case, native multi-class semi-supervised learning are analyzed less intensively. A multi-class S3VM approach was proposed in (Xu and Schuurmans, 2005), using a boosting strategy and employing sparse Newton-optimization. Another recently published multi-class boosting technique in (Tanha et al., 2014) introduces a cost function based on empirical error of labeled data and similarity between labeled and unlabeled data. However, to solve the cost function as a convex problem the employed similarity metric has to be a valid kernel, i.e. psd.

In contrast with the black box property of SVM and its semi-supervised variants, prototype-based methods enjoy a wide popularity in various application domains (Grbovic and Vucetic, 2013; Ortiz-Bayliss et al., 2013) due to their intuitive and simple behavior: they represent their decision in terms of typical representatives (referred to as prototypes) in the input space and classification is based on the distances of data to prototypes. Prototypes can be directly inspected by domain experts in the field in the same way as data points. Popular supervised techniques include standard learning vector quantization (LVQ) and extensions to more powerful settings such as variants based on cost functions such as generalized LVQ (GLVQ) or robust soft LVQ (RSLVQ) (Sato and Yamada, 1995; Seo and Obermayer, 2003), etc.. A recently published prototype-based method extends the ability of GLVQ such that it can directly deal with dissimilarity data (Hammer et al., 2014), which we will use for semi-supervised problems.

In this paper we extend the prototype-based classifier proposed in (Hammer et al., 2014) by self-training approach for semi-supervised tasks employing the conformal prediction technique (Vovk et al., 2005), which provides a confidence measure of the classification. Using the confidence values a so-called *secure region* of unlabeled data can be identified during self-training and used in the retraining. This can potentially enhance the performance of the training, and at the same time conformal prediction estimates a so-called *insecure region* of labeled data helping to adapt the model complexity. This paper is organized as follows. First we give a short review of the prototype-based technique for dissimilarity learning which we will use in the sequel in Section 2. Subsequently, in Section 3, we briefly introduce the concept of conformal prediction. Thereafter we show how to combine both techniques in self-training for semi-supervised learning in Section 4. In Section 5 we show the effectiveness of our technique on simulated data, compare it to state-of-the-art methods on SSL benchmarks, and show results for biomedical dissimilarity data. Finally we summarize our results and discuss potential extensions.

2. Prototype-based relational learning

The basic idea of LVQ is to model data distribution(s) by positioning prototypes in the data space as accurately as possible. Assume data are given as vectors: $\mathbf{x}_i \in \mathbb{R}^d$, $i = 1, \ldots, N$ with label $l_i \in \mathbb{L} = \{1, \ldots, L\}$. LVQ is characterized by m prototypes $\mathbf{w}_j \in \mathbb{R}^d$ in the same space with priorly defined labels $c(\mathbf{w}_j) \in \mathbb{L}$. Besides classic heuristically motivated methods, one of the well-known cost function based learning vector quantization techniques is Generalized LVQ (GLVQ) from (Sato and Yamada, 1995).

Training of GLVQ aims at finding the positions of the prototypes while also taking the generalization ability into account, using the cost function

$$E_{GLVQ} = \sum_{i=1}^{N} \Phi\left(\frac{d(\mathbf{x}_i, \mathbf{w}^+(\mathbf{x}_i)) - d(\mathbf{x}_i, \mathbf{w}^-(\mathbf{x}_i))}{d(\mathbf{x}_i, \mathbf{w}^+(\mathbf{x}_i)) + d(\mathbf{x}_i, \mathbf{w}^-(\mathbf{x}_i))}\right) \tag{1}$$

where $\mathbf{w}^+(\mathbf{x}_i)$ is the closest prototype with the same label as $\mathbf{x}_i$ and $\mathbf{w}^-(\mathbf{x}_i)$ is the closest prototype with a different label than $\mathbf{x}_i$. $d(\cdot, \cdot)$ is the squared Euclidean distance. Φ is a monotonically increasing function, e.g. $\Phi(x) = (1 + \exp(-x))^{-1}$. GLVQ tries to minimize the cost function (1) by means of a stochastic gradient descent, leading to Hebbian learning rules of prototypes, i.e. the closest prototype with the same label is attracted to $\mathbf{x}_i$ while the one with different label is pushed away from $\mathbf{x}_i$. Classification takes place by a so-called "winner takes all" principle: $\mathbf{x} \mapsto c(\mathbf{w}_j)$ where $d(\mathbf{x}, \mathbf{w}_j)$ is minimum, i.e. a new data point is labeled by the closest prototype.

GLVQ models have excellent generalization ability (Biehl et al., 2007), however, they severely depend on the underlying metric, which is usually chosen as Euclidean metric. Recent research has extended GLVQ to directly deal with dissimilarity data (Hammer et al., 2014), which we will discuss in the following.

Let $\mathbf{v}_j \in \mathbb{V}$ be a set of objects, defined in some data space, with $|\mathbb{V}| = N$. We assume, there exists a dissimilarity measure such that $D \in \mathbb{R}^{N \times N}$ is a dissimilarity matrix measuring the pairwise dissimilarities $D_{ij} = d(\mathbf{v}_i, \mathbf{v}_j)$ between all pairs $(\mathbf{v}_i, \mathbf{v}_j) \in \mathbb{V} \times \mathbb{V}$. Any reasonable (possibly non-metric) distance measure is sufficient. Additionally, we assume zero diagonal $d(\mathbf{v}_i, \mathbf{v}_i) = 0$ for all i and symmetry $d(\mathbf{v}_i, \mathbf{v}_j) = d(\mathbf{v}_j, \mathbf{v}_i)$ for all $\{i, j\}$. Thereby, $\mathbf{v}_k$ is represented implicitly by a vector of known dissimilarities with respect to all $\mathbf{v}_j \in \mathbb{V}$. A training set is given where data point $\mathbf{v}_j$ is labeled by $\mathbf{l}_j \in \mathbb{L}$. As detailed in (Pekalska and Duin, 2005), dissimilarity data can always be embedded in pseudo-euclidean space in such a way that $d(\mathbf{v}_i, \mathbf{v}_j)$ is induced by a symmetric (but possibly not positive semi-definite) bilinear form.

For dissimilarity data classification, the key assumption is to restrict prototype positions to linear combinations of data points

of the form

$$\mathbf{w}_j = \sum_i \gamma_{ji}\mathbf{v}_i \text{ with } \sum_i \gamma_{ji} = 1 \tag{2}$$

in the pseudo-Euclidean space. Then dissimilarities between data points and prototypes can be computed implicitly by means of

$$d(\mathbf{v}_i, \mathbf{w}_j) = [D \cdot \gamma_j]_i - \frac{1}{2} \cdot \gamma_j^t D \gamma_j \tag{3}$$

where $\gamma_j = (\gamma_{j1}, \ldots, \gamma_{jn})$ refers to the vector of coefficients describing prototype $\mathbf{w}_j$.

Thus, the cost function of GLVQ (1) can be transferred to the relational setting by substituting distances by (3). This way the cost function of *Relational Generalized Learning Vector Quantization* (RGLVQ) results. In the same way, by means of stochastic gradient descent, the update rules of prototypes can be obtained. The prototypes are initialized as random vectors corresponding to random values γ_{ij} which sum to one. It is possible to take class information into account by setting all γ_{ij} to zero which do not correspond to the class of the prototype. Out-of-sample extension of the classification to new data is possible as follows: For a novel data point $\mathbf{v}$ characterized by its pairwise dissimilarities $D(\mathbf{v})$ to the data used for training, the dissimilarity of $\mathbf{v}$ to a prototype γ_j is $d(\mathbf{v}, \mathbf{w}_j) = D(\mathbf{v})^t \cdot \gamma_j - 1/2 \cdot \gamma_j^t D \gamma_j$, i.e. the data point is assigned to the label of the closest prototype. More details and the generalization ability can be found in (Hammer et al., 2014).

2.1. Limitations

RGLVQ models work very effectively as shown in (Hammer et al., 2014), but they have two major limitations. They are crisp classifiers, where the classification function predicts only the class label but without any additional information about the confidence of the prediction. Especially in the life science some kind of reliability measure, similar to statistical p- or q-values would be beneficial. Only few attempts exist to give reliability estimates for these methods (see e.g. (Cordella et al., 1999; de Stefano et al., 2000)). The second drawback is that the complexity of the model in terms of the number of prototypes needs to be specified a priori. There are some extensions investigated to automatically adjust the number of prototypes by adding new prototypes or deleting redundant ones (e.g. (Grbovic and Vucetic, 2009)), but most of them are restricted to vector space and based on heuristics, but not in a statistical sense. Especially, they can not be directly transferred to dissimilarity data.

In this contribution, we propose to use conformal prediction to enhance classification results with a level of confidence, and to automatically grow a model with suitable model complexity. Reliability, sometimes also referred to as confidence, has been the subject of a theory called *conformal prediction* as introduced in (Vovk et al., 2005). In the next section we will briefly introduce the concept of conformal prediction.

3. Conformal prediction

Conformal prediction is a statistical method assessing each classification decision by providing two measures: *credibility* and *confidence*. Thereby, this technique can be accompanied by a formal stability analysis as provided in (Vovk et al., 2005).

Denote the labeled training data $\mathbf{z}_i = (\mathbf{v}_i, \mathbf{l}_i) \in \mathbb{Z} = \mathbb{V} \times \mathbb{L}$. Furthermore let $\mathbf{v}_{N+1}$ be a new data point with unknown label $\mathbf{l}_{N+1}$, i.e. $\mathbf{z}_{N+1} := (\mathbf{v}_{N+1}, \mathbf{l}_{N+1})$. For given training data $(\mathbf{z}_i)_{i=1,\ldots,N}$, an observed data point $\mathbf{v}_{N+1}$, and a chosen *significance level* ϵ, the *conformal prediction* computes an $(1-\epsilon)$-*prediction region* $\Gamma^\epsilon(\mathbf{z}_1, \ldots, \mathbf{z}_l, \mathbf{v}_{N+1}) \subseteq \mathbb{L}$ consisting of a number of possible label assignments. The applied method ensures that if the data $\mathbf{z}_i$ are *exchangeable*[1] then $P(\mathbf{l}_{N+1} \notin \Gamma^\epsilon(\mathbf{z}_1, \ldots, \mathbf{z}_N, \mathbf{v}_{N+1})) \leq \epsilon$ holds for each distribution of $\mathbb{Z}$. One says that the predictor is *valid*. It is important to mention that the probability is unconditional, such that if we repeat the process of drawing data $(\mathbf{z}_1, ..., \mathbf{z}_N, \mathbf{v}_{N+1})$ and generating Γ^ϵ a number of n times we will find that in at most $\epsilon \cdot n$ cases the real label $\mathbf{l}_{N+1}$ is not among the predicted labels of Γ^ϵ, if statistical fluctuations are ignored.

Prediction region and non-conformity measure

To compute the conformal prediction region Γ^ϵ, a *non-conformity measure* is fixed $A(\mathcal{D}, \mathbf{z})$. It is used to calculate a non-conformity value α that estimates how an observation $\mathbf{z}$ fits to given data $\mathcal{D}=\{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$. In theory, any measure could be used, providing a nontrivial result for suitable choices only. It is the part of the method that can incorporate detailed knowledge about the data distribution. As we focus on prototype-based methods, for a given $\mathbf{z} = (\mathbf{x}, \mathbf{l})$ and a trained relational GLVQ model, we choose as non-conformity measure

$$\alpha_{\mathbf{x}}^{\mathbf{l}} = \frac{d^+(\mathbf{x})}{d^-(\mathbf{x})} \tag{4}$$

with $d^+(\mathbf{x})$ being the distance between $\mathbf{x}$ and the closest prototype labeled $\mathbf{l}$, and $d^-(\mathbf{x})$ being the distance between $\mathbf{x}$ and the closest prototype labeled differently than $\mathbf{l}$ where distances are computed according to Eq. (3). We expect that values $\alpha_x^{\mathbf{l}}$ are small for data $\mathbf{z}$ for which the prediction has high confidence, but it is large if the label does not comply with data. Alternatively, the term in the cost function of GLVQ (1) can also be considered as non-conformity measure.

Given a non-conformity measure A, significance level ϵ, examples $\mathbf{z}_1, \ldots, \mathbf{z}_N$, object $\mathbf{v}_{N+1}$ and a possible label $\mathbf{l}$, it is decided whether $\mathbf{l}$ is contained in $\Gamma^\epsilon(\mathbf{z}_1, \ldots, \mathbf{z}_N, \mathbf{v}_{N+1})$ according to algorithm 1. However, this method would entail high computational costs, especially for large data sets, because this procedure has to be done for all leave-one-out multi-sets for each of the test objects with all possible labels $(\mathbf{v}_{N+1}, \mathbf{l})$. To overcome this problem, some extensions of conformal prediction have been published, i.e. *Inductive Conformal Prediction* (ICP) (Vovk et al., 2005; Vovk, 2012a) and *Cross Conformal Prediction* (CCP) (Vovk, 2012b). Inductive conformal prediction divides the training data into two subsets: *proper training set* and *calibration set*. The model is trained on the proper training set and then used to calculate the non-conformity values of the calibration set. For new data points, classification takes place only based on the non-conformity of the calibration set. As pointed out by (Vovk, 2012a) the size of the calibration set should be reasonably large to cover the data statistic. Although ICP is computationally more efficient, since the training process only

[1] *exchangeability* is a weaker condition than data being i.i.d. which is readily applicable to the online setting as well, for example (Vovk et al., 2005)

Algorithm 1 Conformal Prediction (CP)

function CP($\mathcal{D}$, $\mathbf{v}_{N+1}$, ϵ)
  for all $\mathbf{l} \in \mathbb{L}$ **do**
    $\mathbf{z}_{N+1} := (\mathbf{v}_{N+1}, \mathbf{l})$
    for $i = 1, \ldots, N+1$ **do**
      $\mathcal{D}_i := \{\mathbf{z}_1, \ldots, \mathbf{z}_{N+1}\} \backslash \{\mathbf{z}_i\}$
      $\alpha_i^{\mathbf{l}} := A(\mathcal{D}_i, \mathbf{z}_i)$ ▷ non-conformity of $\mathbf{z}_i$ against $\mathcal{D}_i$
    end for
    $p_{N+1}^{\mathbf{l}} := \frac{|\{i=1,\ldots,N+1 \,|\, \alpha_i^{\mathbf{l}} \geq \alpha_{N+1}^{\mathbf{l}}\}|}{N+1}$
  end for
  return $\Gamma^{\epsilon} := \{\mathbf{l} : p_{N+1}^{\mathbf{l}} > \epsilon\}$
end function

has to be done once, it is predictively less efficient in comparison to the original conformal prediction, in which the training set serves as proper training set and also as calibration set. To avoid this problem another approach, cross-conformal prediction has been proposed, which combines cross-validation with inductive conformal prediction. During the cross-validation process (by taking one fold as calibration set and the remaining folds as proper training set) the data statistic of the whole training set is accumulatively considered, finally the non-conformity of each calibration is merged to classify new data, see (Vovk, 2012b) for more details.

In this work we focus on semi-supervised problems, hence the size of the training set (i.e. labeled data) is usually not large such that we can not use ICP or CCP for our purpose. We decided to modify the original conformal prediction in a different way: we do not match the model exactly against each data set $\mathcal{D}_i$ but instead use the whole training data (i.e. $\mathcal{D}$, excluding $\mathbf{z}_{N+1}$). In this way learning must be performed only once on $\mathcal{D}$. This procedure is motivated by two facts: (1) since we intend to use prototype-based method to train the model, the positions of prototypes depend on the whole data distribution and are in general not widely affected by a single data point, (2) the information loss will be small if the number of training data is reasonably large, so that adding $\mathbf{z}_i$ but leaving out $\mathbf{z}_{N+1}$ will not affect the learning results.

Confidence and credibility

The prediction region $\Gamma^{\epsilon}(\mathbf{z}_1, \ldots, \mathbf{z_N}, \mathbf{v}_{N+1})$ stands in the center of conformal prediction. For a given significance level ϵ it contains the possible labels of $\mathbb{L}$. But how can we use it for prediction?

Suppose we use a meaningful non-conformity measure A, e.g. (4). If the value ϵ is approaching 0, a conformal prediction with almost no errors is required, which can only be satisfied if the prediction region contains all possible labels. If we raise ϵ we allow errors to occur and as a benefit the conformal prediction algorithm excludes unlikely labels from our prediction region, increasing its information content. In detail those $\mathbf{l}$ are discarded for which the p^l-value is less or equal ϵ. Hence only a few $\mathbf{z}_i$ are as non conformal as $\mathbf{z}_{N+1} = (\mathbf{v}_{N+1}, \mathbf{l})$. This is a strong indicator that $\mathbf{z}_{N+1}$ does not belong to the data distribution $\mathbb{Z}$ and so $\mathbf{l}$ does not seem to be the right label. If one further raises ϵ only those $\mathbf{l}$ remain in the conformal region that can produce a high p^l-value meaning that the corresponding $\mathbf{z}_{N+1}$ is rated as very typical by A.

So one can trade significance level against information content. The most useful prediction is that containing exactly one label. Therefore, given an input $\mathbf{v}_i$ two error rates are of particular interest, ϵ_1^i being the smallest ϵ and ϵ_2^i being the largest ϵ so that $|\Gamma^{\epsilon}(\mathcal{D}, \mathbf{v}_i)| = 1$. ϵ_2^i is the p-value of the best and ϵ_1^i is the p-value of the second best label. Thus, typically, a conformal predictor outputs the label $\mathbf{l}$ which describes the prediction region for such choices ϵ, i.e. $\Gamma^{\epsilon} = \{\mathbf{l}\}$, and the classification is accompanied by the two measures

$$\text{confidence} : cf_i := 1 - \epsilon_1^i = 1 - p^{\mathbf{l}_{\text{2nd}}} \tag{5}$$

$$\text{credibility} : cr_i := \epsilon_2^i = p^{\mathbf{l}_{\text{1st}}} \tag{6}$$

Confidence says something about being sure that the second best label and all worse ones are wrong. *Credibility* says something about to be sure that the best label is right respectively that the data point is typical and not an outlier.

The non-conformity measure has a direct impact on the efficiency of the prediction region. A good, informative measure will exclude wrong labels for small error rates and will reject typical data only for high significance levels, meaning that $\epsilon_2^i - \epsilon_1^i$ is large for typical data $\mathbf{v}_i$. That means, that a good measure can give useful information already for low significance level ϵ_1^i and on the other hand one would have to face up a high average significance level ϵ_2^i to exclude the right label from the prediction region.

We would like to point out that the concept of conformal prediction permits pointwise measures of confidence which change if the training data is adapted, also if the decision boundaries remain the same. This means, that similar as in classical statistics, more densely populated training regions permit a better confidence in a decision. Due to the definition of conformal prediction, this is automatically achieved also in online scenarios.

4. Semi-supervised conformal relational GLVQ

RGLVQ opens a way to directly deal with dissimilarity data. As mentioned in section 2.1 it has two major limitations: (i) It is a crisp classifier without any additional information about the confidence of the prediction and (ii) the number of prototypes has to be defined in advance. In the supervised case, these problems have been already addressed by (Schleif et al., 2014, 2009). In (Schleif et al., 2014) the concept of inductive conformal prediction is integrated into a sparse prototype-based classifier for dissimilarity learning problems resulting a sparse prototypical representation of data. In this work we focus on semi-supervised case and by extending our previous work (Zhu et al., 2013b,a) we propose a prototype-based conformal classifier with self-adaptation of model complexity based on the data with high confidence and high credibility values provided by conformal prediction.

First, we denote T_{lab} as labeled data and T_{unlab} as unlabeled data. Generally, in semi-supervised learning unlabeled data are used to improve the trained model based on labeled data in some way. *Self-training* is a very simple approach, which takes iteratively a part of the unlabeled data with predicted labels as new training data into the retraining process to optimize the model, as shown in Algorithm 2. As pointed out by (Zhu and Goldberg, 2009), the key assumption of self-training is that the predictions, at least the high confidence ones, tend to be correct. S should consist of the unlabeled data with the most confident predictions.

Algorithm 2 Self training

1: T_{lab}:= labeled data, T_{unlab}:= unlabeled data
2: **repeat**
3: Train model f based on T_{lab} using supervised learning
4: Apply f to T_{unlab}
5: remove a subset S from T_{unlab} and add $\{(x, f(x))|x \in S\}$ to T_{lab}

In this work we combine conformal prediction with self-training to find the most confident unlabeled data (see Algorithm 3). After random initialization of the model, we train the model on labeled data (T_{lab}) using RGLVQ, based on the model we proceed with the conformal prediction step (line 12-17): For T_{lab} and T_{unlab}, we compute non-conformity values (α) according to (4) (line 12, 13). Based on these non-conformity values a p-value is estimated for each possible label and each unlabeled point from T_{unlab} (line 14, 15). For classification using the conformal classifier, the label of a unlabeled item will be finally predicted as the label with the largest p-value. This refers to the label set provided by the conformal predictor which contains only one label. More complex schemes, by analyzing for example label sets with more than one label would be possible as well, but are not further considered here. The confidence value (cf_i) is given as one minus the second largest p-value (eq. (5)) and the credibility (cr_i) is the largest p-value of this item (eq. (6)) (line 16, 17).

Data used for self-training

In order to identify unlabeled items with high confidence predictions we define a measure cc as the product of confidence and credibility values: For a given data point $\mathbf{v}_i \in T_{\text{unlab}}$,

$$cc_i := cf_i \cdot cr_i \tag{7}$$

Actually, any other reasonable indicator can be used here which can detect the high confidence and high credibility values at the same time. In this case the sum of both values is not appropriate, since one of them can dominate the sum. A high cc-value of a unlabeled item indicates that with high probability its predicted label (that with the highest p-value) is the true underlying label.

For self-training the unlabeled data with predicted labels of high probability can be taken into the next retraining. The region which consists of these unlabeled items is referred to as '*secure region*' (denoted as $\mathcal{SR}$). To identify $\mathcal{SR}$ we take a fraction (prc) of the top cc-values of the unlabeled data[2].

Adaptation of model complexity

On the other hand we also collect a set of points of the "labeled" data (i.e. original labeled items and the items with high cc-values labeled by previous iterations) with low credibility and confidence values, which builds a so-called '*insecure region*' ($\mathcal{ISR}$) of the training data,

$$\mathcal{ISR} := \{\mathbf{v}_i \in T_{\text{lab}} : cf_i \leq \zeta_1 \vee cr_i \leq \zeta_2\}. \tag{8}$$

A low confidence value is given if the confidence value cf_i or the credibility cr_i below a user defined threshold ζ_i or ζ_2, respectively. Defined values for ζ_1 or ζ_2 can be derived from the quantiles of confidence/credibilty values as observed in the data. The $\mathcal{ISR}$ will be represented by a new prototype as the "median" of $\mathcal{ISR}$. Here the notion "median" refers to the values cf_i and cr_i in this context. For both, we determine the (one dimensional) median cf_m and $cr_{m'}$, respectively, and represent the set $\mathcal{ISR}$ by one (if $m = m'$) or two (if $m \neq m'$) exemplars which cause these values. This step automatically adapts the complexity of the model, i.e. the number of prototypes. In the retraining this new prototype will be also trained on the new training data.

During the self-training process the training set T_{lab} is iteratively augmented by adding the secure region of the unlabeled data $\mathcal{SR}$ to itself while the unlabeled data T_{unlab} is shrunk by discarding the secure region. The performance of the retraining is evaluated based on the original labeled data only (EvalSet = T_{lab}). The method terminates if the improvement of the performance is not significant (less than 1%) after a certain number of iterations ($win_{\text{max\_itr}}$) or the maximal number of iterations are reached (max_{itr}) or the insecure region ($\mathcal{ISR}$) is too small or the unlabeled set T_{unlab} is empty, i.e. all unlabeled data have been considered in the retraining. The proposed method is referred to as *Secure Semi-Supervised Conformal RGLVQ* (S3-C-RGLVQ).

Computational complexity of S3-C-RGLVQ

The runtime complexity of original RGLVQ is quadratic with respect to the size of the training data, i.e. $O(N^2)$, since the whole dissimilarity matrix has to be deal with. For S3-C-RGLVQ, assuming that $N = |T_{\text{lab}}| + |T_{\text{unlab}}|$, the complexity of RGLVQ is decreased to $O(|T_{\text{lab}}|^2)$. Due to the model adaptation S3-C-RGLVQ has to retrain the model k times and the size of labeled data is increased (but at most $|T_{\text{lab}}| = N$), so the retraining process of S3-C-RGLVQ can not get rid of $O(N^2 \cdot k)$, normally $k \ll N$, thus the complexity remains $O(N^2)$. Additionally, the runtime complexity of conformal perdition step can be considered as linear $O(N)$, since for each retraining the α-values for unlabeled data with respect to all possible labels have to be calculated, i.e. $O(k \cdot |T_{\text{unlab}}| \cdot |\mathbb{L}|)$, usually for semi-supervised problems $T_{\text{unlab}} \approx N$, and normally $|\mathbb{L}| \ll N$, so this step stays linear. Overall, we get $O(N^2)$ for retraining step and $O(N)$ for conformal prediction step.

5. Experiments

We evaluate S3-C-RGLVQ on a large range of tasks. First, we demonstrate its performance for two artificial data sets: checkerboard data and banana-shaped data, with known vector representation to show the ability of dealing with partially labeled data, especially non i.i.d labeled data. Then we compare S3-C-RGLVQ with state-of-the-art semi-supervised SVMs on SSL binary-class benchmarks. For vectorial data the dissimilarity matrices D are obtained using the squared-Euclidean distance. Additionally, five real life non-vectorial multi-class data sets from the bioinformatics domain are used to compare with original RGLVQ (trained only on labeled data).

Artificial data sets: The checkerboard data set consists of two classes with 1200 data points, in two dimensions and $2 \cdot 2$ clusters. The clusters with different classes distribute along

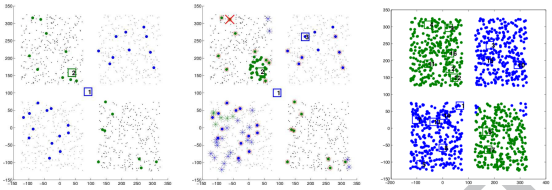
[2] prc is customizable and in our experiments we set prc = 5% which is a good compromise between learning performance and efficiency.

Algorithm 3 secure semi-supervised conformal RGLVQ

init: W : randomly initialized, $W_{\text{new}} := \emptyset$, $W_{\text{best}} := W$, $\mathcal{ISR} := \emptyset$; $\mathcal{SR} := \emptyset$; EvalSet = T_{lab}; $ctn_{\text{best}} = 0$; $max_{\text{itr}} = 100$; $win_{\text{max\_itr}} = 10$; $acc_{\text{best}} = 0$
repeat ▷ self-training process
  $W := W \bigcup W_{\text{new}}$
  $T_{\text{lab}} := T_{\text{lab}} \cup \mathcal{SR}$, $T_{\text{unlab}} := T_{\text{unlab}} \backslash \mathcal{SR}$
  $W :=$ train T_{lab} by RGLVQ given W
  $acc :=$ evaluation of W on $EvalSet$;
  if $acc - acc_{\text{best}} \geq 1\%$ **then**
    $W_{\text{best}} = W$, $acc_{\text{best}} = acc$, $ctn_{\text{best}} = 0$
  else
    $ctn_{\text{best}} = ctn_{\text{best}} + 1$
  end if
  $A_{T_{\text{lab}}} := \{\alpha_i \mid i \in T_{\text{lab}}\}$ ▷ conformal prediction step
  $A^{\mathbb{L}}_{T_{\text{unlab}}} := \{\alpha_i^l \mid i \in T_{\text{unlab}},\ l \in \mathbb{L}\}$
  $P_{T_{\text{lab}}} := \{p_i \mid i \in T_{\text{lab}}\}$
  $P^{\mathbb{L}}_{T_{\text{unlab}}} := \{p_i^l \mid i \in T_{\text{unlab}},\ l \in \mathbb{L}\}$
  $CF_{T_{\text{lab}}} := \{cf_i \mid i \in T_{\text{lab}}\}$; $CR_{T_{\text{lab}}} := \{cr_i \mid i \in T_{\text{lab}}\}$;
  $CF_{T_{\text{unlab}}} := \{cf_i \mid i \in T_{\text{unlab}}\}$; $CR_{T_{\text{unlab}}} := \{cr_i \mid i \in T_{\text{unlab}}\}$;
  generate $\mathcal{SR}$ of T_{unlab} based on $CF_{T_{\text{unlab}}}$ and $CR_{T_{\text{unlab}}}$
  generate $\mathcal{ISR}$ of T_{lab} based on $CF_{T_{\text{lab}}}$ and $CR_{T_{\text{lab}}}$
  generate W_{new} from $\mathcal{ISR}$ ▷ new prototype(s)
  $itr = itr + 1$
until $|\mathcal{ISR}| < 1\% \cdot |T_{\text{lab}}|$ or $itr = max_{itr}$ or $ctn_{\text{best}} = win_{\text{max\_itr}}$ or $T_{\text{unlab}} = \emptyset$
return W_{best};

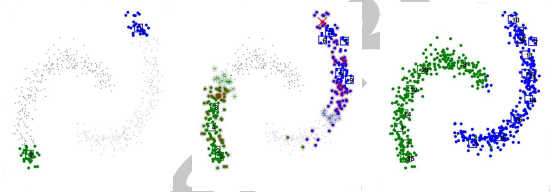
each axis. We randomly select about 3% as labeled data and the remaining data as unlabeled data. The prototypes are randomly initialized based on labeled data and one prototype per class. RGLVQ can learn these data only if the prototypes are initialized near the centers of the multi-modal distributions, provided a sufficient number of prototypes. The S3-C-RGLVQ on the other hand automatically adapts its model complexity according to the introduced scheme, leading to an effective model with minimum initialization of one prototype per class only. As an example, Figure 1 shows some intermediate results up to convergence. We randomly initialized two prototypes only on labeled data. Figure 1(a) shows that after the initial training two prototypes (marked by squares) are located in the center of the labeled data. Obviously, in this case one prototype per each class is not sufficient to model the whole data space. In Figure 1(b) after the conformal prediction process, the secure region of unlabeled data (marked by stars) and the insecure region of labeled data (marked by red circles) can be identified. To 'cover' the insecure region a new prototype (marked by red cross) is added right there.

Moreover, there are some unlabeled data misclassified by CP, which will be taken into the current retraining process. The reason thereof is that due to the smaller number of prototypes at the early stage which are not well distributed into the multi-modal clusters, a reasonable number of points with relatively lower confidence/credibility values (i.e. lower cc-value) exists, which is a natural consequence, because by chance 50% got the correct label. By a larger value of the parameter 'prc' some of these points can be considered in the next training. In this case those points can also be considered as outliers. Due to the fact that the prototype-based method is very stable against outliers, i.e. the positions of prototypes depend on the whole data distribution and are not widely affected by a single point, the movement of the prototypes is mainly dominated by the correctly classified points and the labeled data. As shown in Figure 1(c), once the algorithm converges, those points can be correctly assigned to their closest prototypes.



(a) Initial training (b) 3. iteration (c) Final model

Fig. 1: Checkerboard data set



(a) Initial training (b) 10. iteration (c) Final model

Fig. 2: Banana-shaped data

Another simulated data set consists of two banana-shaped data clouds indicating two classes. Each banana consists of 300 two dimensional data points in Figure 2. We randomly select non i.i.d. a small fraction (ca. 5%) of each banana as labeled data, the remaining as unlabeled data. With the same setting for checkerboard data we start with one prototype per class and train the initial model on the labeled data as shown in Fig. 2(a). The number of prototypes increased step-wise during the retraining process by adding new prototype in the insecure region, while by means of secure region the unlabeled data are iteratively considered. Thereby at the end the data manifold can be well studied.

UCI two-class data sets: Furthermore, we evaluate the proposed method on different widely used benchmarks for semi-supervised learning from the UCI repository[3] and compare it with the best semi-supervised SVM with RBF-kernel taken from (Li and Zhou, 2011)[4]. To keep the same experimental setting, we randomly select 100 examples of the data to be used as labeled examples, and use the remaining data as unlabeled data. For initial prototypes, we use the same setting as previous experiments: one randomly initialized prototype for each class. The experiments are repeated for 12 times and the average test-set accuracy (on the unlabeled data) and standard deviation are reported in Table 1. Except voting data, the proposed method provides comparable results for all remaining data sets.

Real life multi-class data sets: Moreover, we also evaluate the methods on five real-life relational data sets from the bioinformatics domain, where no direct vector embedding exists and the data are given as (dis-)similarities. These data sets constitute typical examples of non-Euclidean data which occur in complex systems, such as medical image analysis, mass spec-

[3] http://archive.ics.uci.edu/ml/datasets.html

[4] In this paper the authors made a comprehensive comparison between different semi-supervised SVMs, e.g. TSVM, S3VM, S4VM, etc. with linear and rbf kernels. For our experiments we pick the best result of rbf-kernel as reference for each data set.

Table 1: Classification accuracy (% ± std) of UCI Benchmarks for two classes problems for SSL

two-class UCI data	S3-C-RGLVQ	Semi-SVM^{best}(rbf)
diabetes	70.17 ± 2.32	70.3 ± 2.1
german	71.61 ± 1.14	71.0 ± 1.1
haberman	**73.30 ± 5.02**	68.3 ± 2.8
voting	89.20 ± 0.89	**92.6 ± 1.6**
wdbc	92.34 ± 1.19	**93.6 ± 1.7**
austrailian	**83.22 ± 1.51**	81.8 ± 1.9
breast-cancer	**96.20 ± 0.51**	95.5 ± 1.0

trometry, and symbolic domains. In all cases, dedicated preprocessing steps and dissimilarity measures for structures are used. The dissimilarity measures are inherently non-Euclidean and cannot be embedded isometrically in a Euclidean vector space.

- The *SwissProt* data consists of 5,791 samples in 10 classes, a subset of the SwissProt database (Boeckmann B, 2003) (release 37). Sequence scores are obtained by the Smith-Waterman algorithm (Gusfield, 1997).
- The *Copenhagen Chromosomes* data consists of 4,200 samples in 20 classes of string representations compared by an edit distance measure (Neuhaus and Bunke, 2006).
- The *Sonatas* data are compression distance scores of 1068 midi files in 5 classes, for details see Hammer et al. (2014).
- The *Zongker* digit dissimilarity data (2000 samples in 10 classes) is taken from (Duin, 2012).
- The Vibrio data set consists of 1,100 samples in 49 classes of bacteria measurements. Details are given in Schleif et al. (2014).

In general, we use the same experimental setting as for the UCI data, but besides random initialization of prototypes other initialization strategies can also be adopted, e.g. affinity propagation (Frey and Dueck, 2007) or relational neural gas (Hammer and Hasenfuss, 2010) which can deal with dissimilarity data. We use relational neural gas (RNG) (class-wise as well as global) to initialize prototypes. For comparison, we report the results of RGLVQ trained only on labeled data to tackle another problem for SSL, i.e. the degeneration issue as discussed by (Singh et al., 2008; Zhu and Goldberg, 2009). In order to keep the comparisons fair the number of prototypes for each class for RGLVQ is set to the number of prototypes for each class of the final S3-C-RGLVQ model. The results are reported in Tables 2 and 3.

In Table 2, for random initialization, in all cases but one, a better classification accuracy can be obtained using conformal prediction compared to original RGLVQ only based on labeled data without consideration of additional information about unlabeled data. The chromosome is a perfectly balanced data set, it leads to the fact that the initial model based only on the labeled data is almost perfectly trained by RGLVQ, so that the potential to improve the model by considering unlabeled information in this case is very limited. Interestingly, a global clustering does significantly increase the classification error, which can be attributed to the fact that the (unsupervised) cluster structure and the interesting classes do not correlate perfectly. A class-wise clustering can improve the accuracy in some cases (swissprot, chromosomes) since the cluster structure of each class can be better studied and this can benefit the training. In

Table 3: Classification accuracy (% ± std) for real life data with two prototypes per class initialized randomly

Initialization	two prototypes per class *random*	
Data	S3-C-RGLVQ	RGLVQ
swissprot	**87.55 ± 2.74**	86.74 ± 2.26
chromosome	85.07 ± 1.67	84.90 ± 1.45
sonatas	72.59 ± 3.83	71.95 ± 2.39
zongker	89.39 ± 1.01	89.59 ± 0.86
vibrio	98.80 ± 0.96	98.79 ± 0.82

Table 3, we start with two randomly initialized prototypes for each class. Comparing to the case with one prototype per class, for some data sets (Sanatas, Vibrio) one prototype per class is sufficient to cover the data space, twice this number did not really improve the learning ability. For the remaining data sets although the accuracies have been significantly improved, the price thereof is paid by a doubled model complexity, indicating that a random initialization with one prototype per class seems a reasonable choice if no further information is available.

In all cases, the incorporation of information about unlabeled data into the classifier leads to an increased, at least equal, classification accuracy of the resulting model, since the additionally available information can better be taken into account to optimize the class boundaries. Thus, S3-C-RGLVQ constitutes a very promising method to infer a high quality semi-supervised prototype-based classifier for general dissimilarity data sets which offers point-wise measures for confidence and credibility about the classification.

6. Conclusions

In this contribution, we have developed an efficient semi-supervised classification technique for general dissimilarity data based on the conformal prediction concept and relational prototype-based classifier. It naturally inherits the merits from both techniques. Due to a prototypical representation, unlike many alternative black-box techniques, it offers the possibility of a direct inspection of the classifier by humans. This technique does not require that data are embeddable into Euclidean space, rather, a general symmetric dissimilarity matrix is sufficient. Due to the properties of conformal prediction, instead of providing only a predicted label, it also permits to identify the safety of the prediction by means of point-wise measures for confidence and credibility. Thereby the 'secure' unlabeled data can be exploited and used to optimize the trained model, at the same time the 'insecure' training data can be identified and accordingly the complexity of the model is adapted.

We demonstrated the quality of the technique on different SSL data sets. As a result, a powerful semi-supervised learning algorithm can be derived, which in most cases achieves comparable results to semi-supervised SVM and with direct interpretability of the classification in term of the prototypes. It works especially well for non i.i.d labeled data. Duo to the multi-class capability of prototype-based methods, it can directly deal with multi-class data sets. Furthermore, it does not degenerate the learning performance by incorporating additional information of unlabeled data which is still a crucial issue in the semi-supervised learning (Singh et al., 2008; Zhu and Goldberg, 2009).

Table 2: Classification accuracy (% ± std) for real life data with one initial prototype per class which are initialized by different strategies: randomly, RNG, and classwise RNG

Initialization	one prototype per class			
	random		*RNG*	*class-wise RNG*
Data	S3-C-RGLVQ	RGLVQ	S3-C-RGLVQ	S3-C-RGLVQ
swissprot	81.06 ± 5.53	79.37 ± 4.78	57.84 ± 6.79	**90.66 ± 2.73**
chromosome	78.88 ± 3.28	78.78 ± 3.70	63.12 ± 5.38	**84.43** ± 1.86
sonatas	**77.98 ± 3.94**	71.99 ± 2.92	69.37 ± 2.24	67.04 ± 2.07
zongker	87.93 ± 0.84	86.48 ± 1.50	**89.25 ± 1.09**	78.11 ± 8.13
vibrio	**98.76 ± 0.47**	97.40 ± 0.84	37.76 ± 4.54	90.16 ± 1.15

One central problem of this technique as introduced above has not yet been considered in this letter: we used a global value *prc* to identify the secure region of the training data in every iteration. It may cause some uncertainty issues at the earlier stages of retraining as we have seen in the checkerboard data, if the number of prototypes is not sufficiently high and the prototypes are not well distributed in the data space. In spite of the fact that this potential issue can be partially solved by the nature of prototype-based method, i.e. its stability against outliers, it should be more seriously studied, e.g. using a local value *prc* for each iteration to more precisely identify the high confidence items. Future work will also address the model sparsity for large scale problem and linear approximation techniques as introduced in (Zhu et al., 2012).

Acknowledgments

This research and development project is funded by the German Federal Ministry of Education and Research (BMBF) within the Leading-Edge Cluster Competition and managed by the Project Management Agency Karlsruhe (PTKA). The authors are responsible for the contents of this publication. Funding in the frame of the centre of excellence 'Cognitive Interaction Technologies' (CITEC) and a Marie Curie Intra-European Fellowship (IEF) FP7-PEOPLE-2012-IEF (FP7-327791-ProMoS) are gratefully acknowledged.

References

Biehl, M., Ghosh, A., Hammer, B., 2007. Dynamics and generalization ability of lvq algorithms. Journal of Machine Learning Research 8, 323–360.

Boeckmann B, e., 2003. The swiss-prot protein knowledgebase and its supplement trembl. Nucleic Acids Research 31, 365–370.

Chapelle, O., Schölkopf, B., Zien, A. (Eds.), 2006. Semi-Supervised Learning. MIT Press, Cambridge, MA.

Cordella, L.P., Foggia, P., Sansone, C., Tortorella, F., Vento, M., 1999. Reliability parameters to improve combination strategies in multi-expert systems. Pattern Analysis and Applications 2, 205–214.

Duin, R.P., 2012. PRTools. URL: `http://www.prtools.org`.

Frey, B.J., Dueck, D., 2007. Clustering by passing messages between data points. Science 315, 972–976.

Grbovic, M., Vucetic, S., 2009. Learning vector quantization with adaptive prototype addition and removal, in: Neural Networks, 2009. IJCNN 2009. International Joint Conference on, pp. 994–1001.

Grbovic, M., Vucetic, S., 2013. Decentralized estimation using distortion sensitive learning vector quantization. Pattern Recognition Letters 34, 963–969.

Gusfield, D., 1997. Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. Cambridge University Press.

Haasdonk, B., Bahlmann, C., 2004. Learning with distance substitution kernels. Pattern Recognition - Proc. of the 26th DAGM Symposium .

Hammer, B., Hasenfuss, A., 2010. Topographic mapping of large dissimilarity datasets. Neural Computation 22, 2229–2284.

Hammer, B., Hofmann, D., Schleif, F.M., Zhu, X., 2014. Learning vector quantization for (dis-)similarities. Neurocomputing 131, 43–51.

Li, Y.F., Zhou, Z.H., 2011. Towards making unlabeled data never hurt, in: Getoor, L., Scheffer, T. (Eds.), ICML, Omnipress. pp. 1081–1088.

Neuhaus, M., Bunke, H., 2006. Edit distance based kernel functions for structural pattern classification. Pattern Recognition 39, 1852–1863.

Ortiz-Bayliss, J., Terashima-Marín, H., Conant-Pablos, S., 2013. Learning vector quantization for variable ordering in constraint satisfaction problems. Pattern Recognition Letters 34, 423–432.

Pekalska, E., Duin, R., 2005. The dissimilarity representation for pattern recognition. World Scientific.

Sato, A., Yamada, K., 1995. Generalized learning vector quantization, in: Touretzky, D.S., Mozer, M., Hasselmo, M.E. (Eds.), NIPS, MIT Press. pp. 423–429.

Schleif, F.M., Ongyerth, F.M., Villmann, T., 2009. Supervised data analysis and reliability estimation for spectral data. NeuroComputing 72, 3590–3601.

Schleif, F.M., Zhu, X., Hammer, B., 2014. Sparse conformal prediction for dissimilarity data. Annals of Mathematics and Artificial Intelligence (AMAI) , 1–22doi:`10.1007/s10472-014-9402-1`.

Seo, S., Obermayer, K., 2003. Soft learning vector quantization. Neural Computation 15, 1589–1604.

Singh, A., Nowak, R.D., Zhu, X., 2008. Unlabeled data: Now it helps, now it doesn't, in: Koller, D., Schuurmans, D., Bengio, Y., Bottou, L. (Eds.), NIPS, Curran Associates, Inc.. pp. 1513–1520.

de Stefano, C., Sansone, C., Vento, M., 2000. To reject or not to reject: that is the question: an answer in case of neural classifiers. IEEE Transactions on Systems, Man and Cybernetics Part C 30, 84–93.

Subramanya, A., Bilmes, J., 2011. Semi-supervised learning with measure propagation. Journal of Machine Learning Research 12, 3311–3370.

Suzuki, J., Fujino, A., Isozaki, H., 2007. Semi-supervised structured output learning based on a hybrid generative and discriminative approach, in: EMNLP-CoNLL, ACL. pp. 791–800.

Tanha, J., van Someren, M., Afsarmanesh, H., 2014. Boosting for multiclass semi-supervised learning. Pattern Recognition Letters 37, 63 – 77.

Tian, X., Gasso, G., Canu, S., 2012. A multiple kernel framework for inductive semi-supervised svm learning. Neurocomputing 90, 46–58.

Trosset, M.W., Priebe, C.E., Park, Y., Miller, M.I., 2008. Semisupervised learning from dissimilarity data. Computational Statistics and Data Analysis 52, 4643 – 4657. doi:`10.1016/j.csda.2008.02.030`.

Vovk, V., 2012a. Conditional validity of inductive conformal predictors. Journal of Machine Learning Research - Proceedings Track 25, 475–490.

Vovk, V., 2012b. Cross-conformal predictors. CoRR abs/1208.0806.

Vovk, V., Gammerman, A., Shafer, G., 2005. Algorithmic Learning in a Random World. Springer, New York.

Xu, L., Schuurmans, D., 2005. Unsupervised and semi-supervised multi-class support vector machines, in: Veloso, M.M., Kambhampati, S. (Eds.), AAAI, AAAI Press / The MIT Press. pp. 904–910.

Zhang, K., Wang, Q., Lan, L., Sun, Y., Marsic, I., 2014. Sparse semi-supervised learning on low-rank kernel. Neurocomputing 129, 265 – 272.

Zhu, X., Gisbrecht, A., Schleif, F.M., Hammer, B., 2012. Approximation techniques for clustering dissimilarity data. Neurocomputing 90, 72–84.

Zhu, X., Goldberg, A.B., 2009. Introduction to semi-supervised learning. Synthesis Lectures on Artif. Intell. and Machine Learning 3, 1–130.

Zhu, X., Schleif, F.M., Hammer, B., 2013a. Secure semi-supervised vector quantization for dissimilarity data, Springer. pp. 347–356.

Zhu, X., Schleif, F.M., Hammer, B., 2013b. Semi-supervised vector quantization for proximity data, in: ESANN, pp. 89–94.